**Experiment 6**

**Aim** - Use burp proxy to test web applications

**Theory**- **Burp Suite** is a comprehensive platform for testing the security of web applications, and one of its core tools is **Burp Proxy**. Burp Proxy allows you to intercept, modify, and analyze all HTTP(S) traffic between your browser and the web server. It is an essential tool for web application security testing, enabling testers to find vulnerabilities like SQL injection, cross-site scripting (XSS), insecure session handling, and more.

### **What is Burp Proxy?**

**Burp Proxy** is a component of Burp Suite that sits between the client (browser) and the server, intercepting all HTTP(S) requests and responses. This makes it possible to:

* **Inspect and manipulate** the raw data sent between the client and server.
* Modify request headers, cookies, or parameters to test how the server responds.
* Identify security flaws, including broken access controls, injection vulnerabilities, and session mismanagement.

The Burp Proxy tool is often used in **manual penetration testing** of web applications to understand how the application handles data and responds to malicious inputs.

### **Setting Up Burp Proxy**

#### **1. Install Burp Suite**

* Download and install Burp Suite from the official website (PortSwigger).
* There are **Community** (free) and **Professional** (paid) versions. The Community edition has limited features but includes Burp Proxy.

#### **2. Configure Browser to Use Burp Proxy**

* You need to configure your browser to route all HTTP(S) traffic through Burp.
* By default, Burp listens on 127.0.0.1:8080. You can adjust the proxy settings in your browser (e.g., Firefox or Chrome) to point to Burp's IP address and port (localhost:8080).
* In most cases, you also need to install Burp’s **CA certificate** to intercept HTTPS traffic without receiving browser warnings. This allows Burp to decrypt HTTPS traffic.

#### **3. Configure SSL/TLS Interception (HTTPS Traffic)**

* Open Burp Suite, go to **Proxy > Intercept** and turn the intercept on.
* Install Burp’s **CA certificate** into your browser so you can intercept HTTPS traffic. Without this, browsers will flag security warnings or block HTTPS connections.
* Burp dynamically generates certificates for each site based on its own CA.

### **How to Use Burp Proxy for Web Application Testing**

Once Burp Proxy is set up, you can begin intercepting and manipulating traffic to test the security of the web application. Here’s an overview of how you might use it:

#### **1. Intercepting HTTP(S) Requests and Responses**

* Open **Proxy > Intercept** and ensure the intercept option is "on."
* Browse the web application normally in your browser.
* Burp will capture and display the HTTP requests and responses in real time, allowing you to see the raw data exchanged between your browser and the server.
* You can modify requests, change headers, tweak parameters, or even add malicious input before forwarding them to the server.

#### **2. Testing for Vulnerabilities**

* **Parameter Tampering**: Modify GET or POST parameters to check for insecure data handling. For example, change numeric IDs in URLs to check for broken access control vulnerabilities (e.g., trying to access other users' data by manipulating IDs).
* **Session Management Testing**: Analyze session cookies, and try to hijack or reuse sessions to see if the web app is vulnerable to session fixation or session ID leakage.
* **Injection Attacks**: Modify form inputs or URL parameters to test for SQL injection, command injection, or NoSQL injection vulnerabilities.
* **Cross-Site Scripting (XSS)**: Inject JavaScript code in input fields or URLs and see how the application renders or executes this code.

#### **3. Repeating Requests and Automating Tests**

* **Repeater Tool**: Burp Proxy integrates with Burp’s **Repeater** tool, allowing you to manually modify and resend requests multiple times. This is useful for refining and testing different payloads.
* **Intruder Tool**: For automated testing of multiple parameters, Burp’s **Intruder** tool can be used in conjunction with Proxy. Intruder sends repeated HTTP requests with slight variations (e.g., payloads) to find vulnerabilities like brute-force login attacks or parameter fuzzing.

#### **4. Analyzing Application Responses**

* Observe how the web application responds to different manipulated requests.
* Check for unusual behaviors, such as error messages that may reveal sensitive information (e.g., database errors indicating SQL injection).
* Pay attention to response headers, such as security headers (e.g., X-Frame-Options, Content-Security-Policy), which indicate the level of security in the web application’s configuration.

### **Common Vulnerabilities Detected with Burp Proxy**

Using Burp Proxy effectively allows you to identify many security vulnerabilities, including:

1. **SQL Injection**: Injecting SQL commands into an application's database via input fields or URL parameters.
   * Modify form inputs to try injecting SQL commands and see how the application responds.
   * Look for error messages or abnormal behavior that indicate database manipulation.
2. **Cross-Site Scripting (XSS)**: Injecting malicious JavaScript into web pages to hijack sessions, redirect users, or perform other actions.
   * Enter JavaScript code into any input fields (e.g., search bars, comment sections) and check if it gets executed.
3. **Cross-Site Request Forgery (CSRF)**: Sending unauthorized commands from an authenticated user.
   * Use intercepted POST requests and replay them to see if the server processes them without proper authentication tokens.
4. **Broken Authentication**: Testing for weak authentication mechanisms, session hijacking, and insecure session handling.
   * Modify cookies, session tokens, and login credentials to check for weaknesses in session management.
5. **Insecure Direct Object References (IDOR)**: Changing object IDs in URLs to access unauthorized resources.
   * Manipulate user IDs or resource IDs in GET/POST requests to see if unauthorized access is possible.
6. **Security Misconfigurations**: Identifying misconfigurations like unnecessary services enabled, open ports, or default credentials.
   * Analyze HTTP headers and server responses to detect misconfigurations.

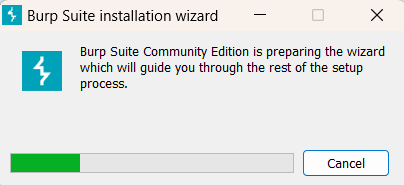
### 

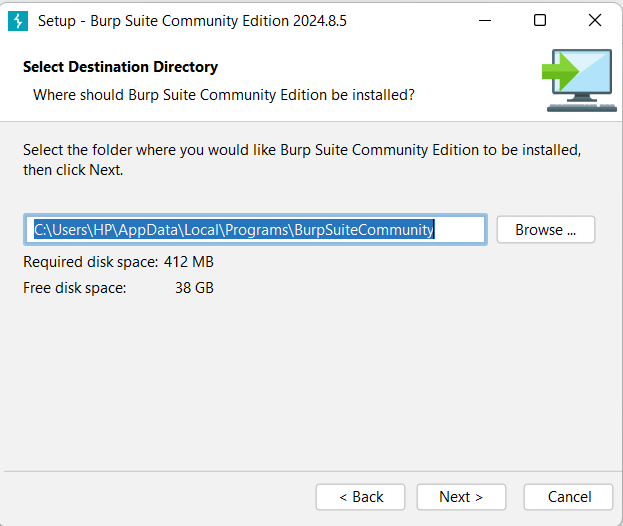
### 

### **Advantages of Using Burp Proxy**

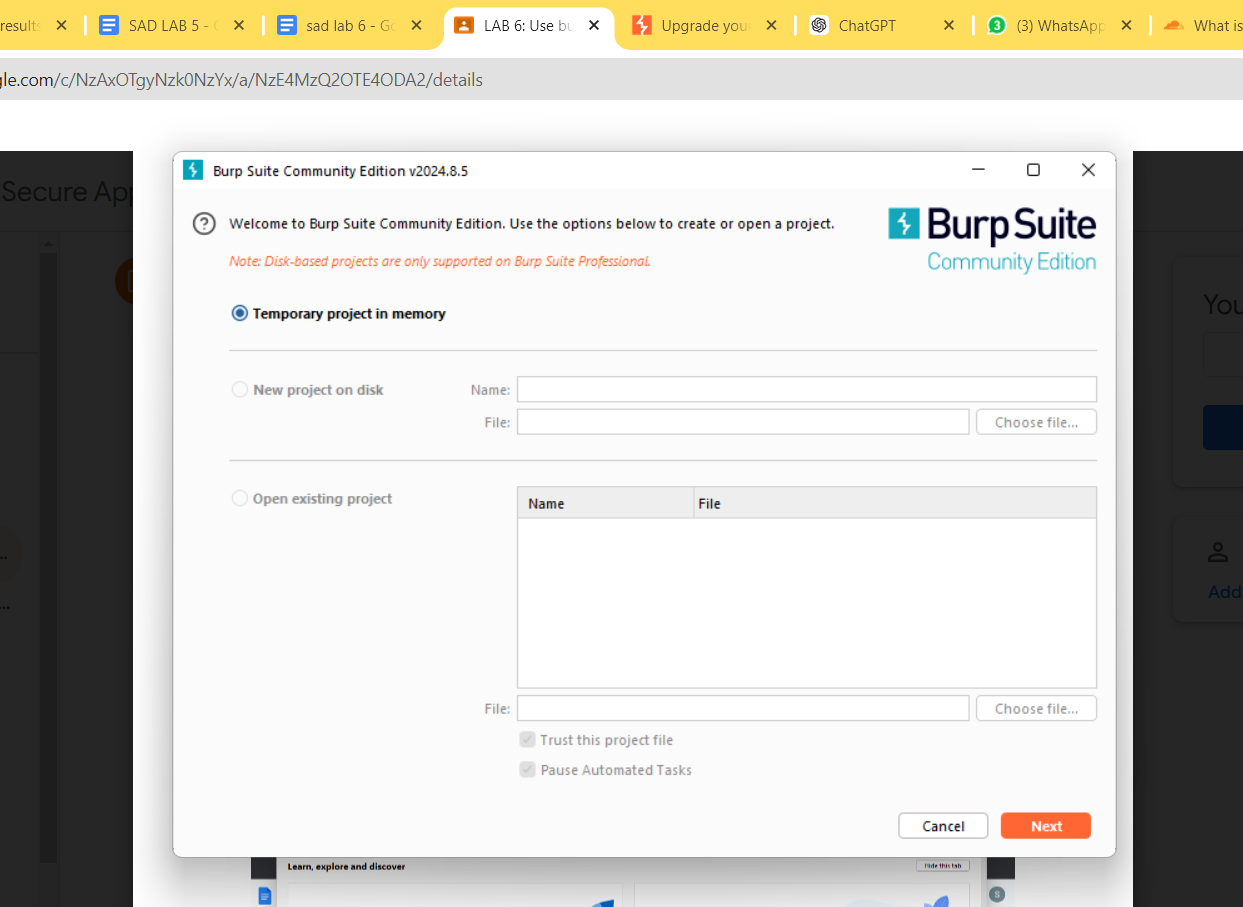
* **Real-time inspection and modification**: Allows you to analyze requests and responses dynamically, making it easier to identify and exploit potential vulnerabilities.
* **Customization and flexibility**: You can easily modify and manipulate traffic to simulate attacks or test security controls.
* **Comprehensive integration**: Works seamlessly with other Burp tools like Repeater, Scanner, and Intruder to facilitate both manual and automated testing.
* **HTTPS support**: Burp can decrypt and inspect HTTPS traffic, enabling security testing of encrypted data exchanges.

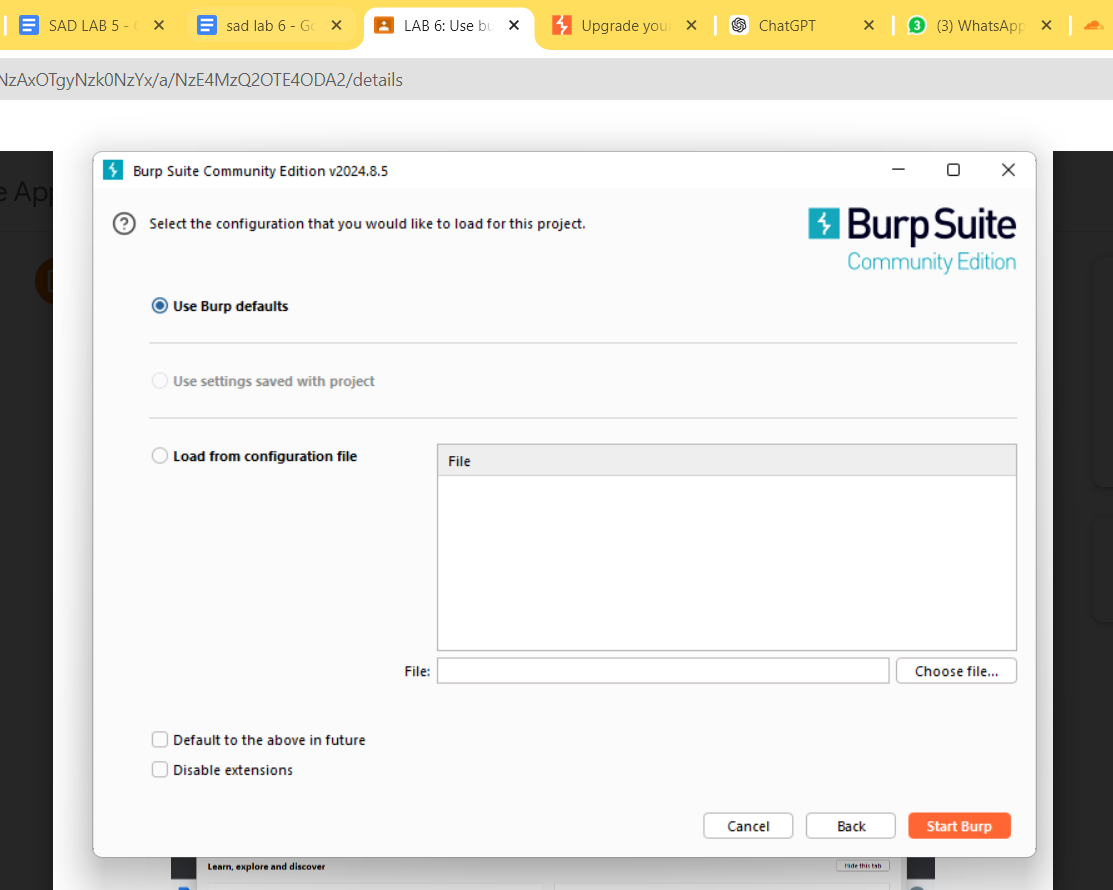
**Step 1: Install Burp Suite**



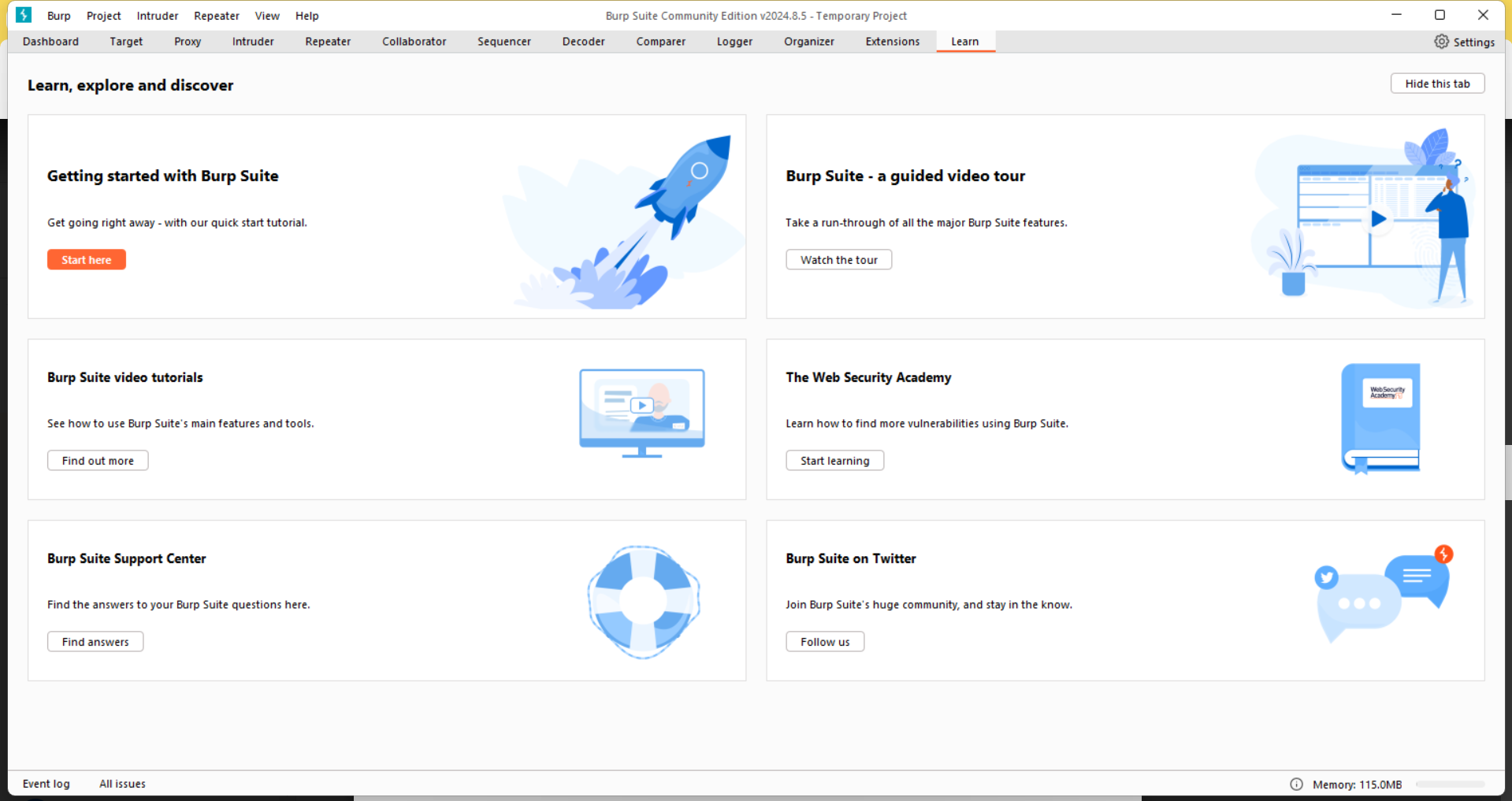


**Step 2 : Create a new temporary project**

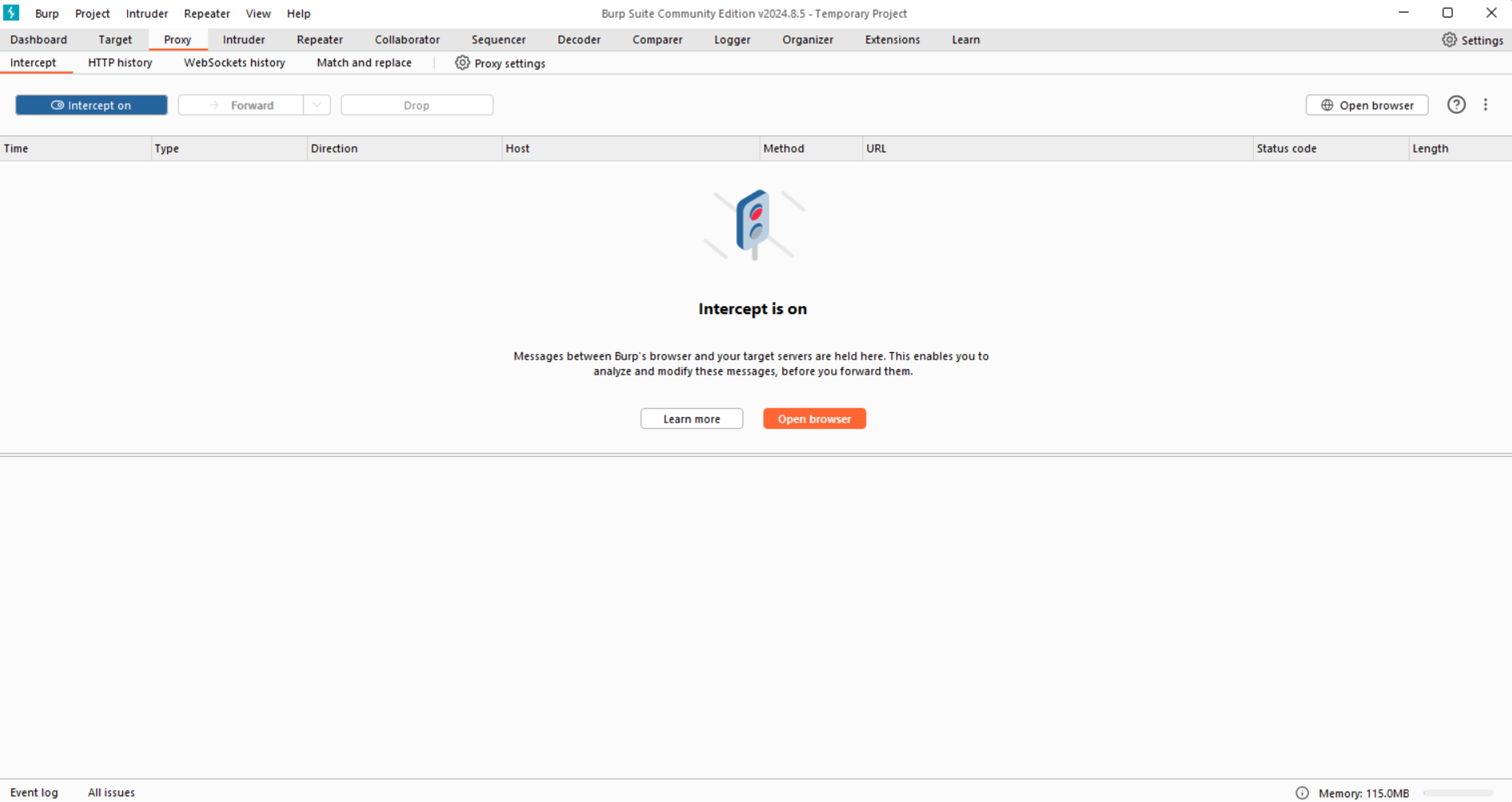




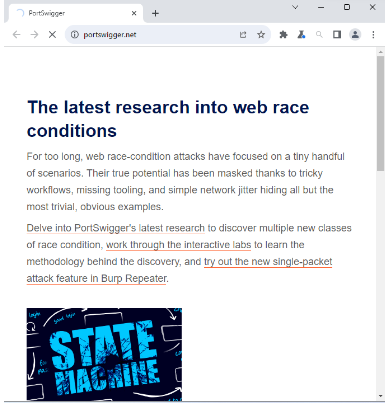
**Step 3 : Click on dashboard to see all statistics**



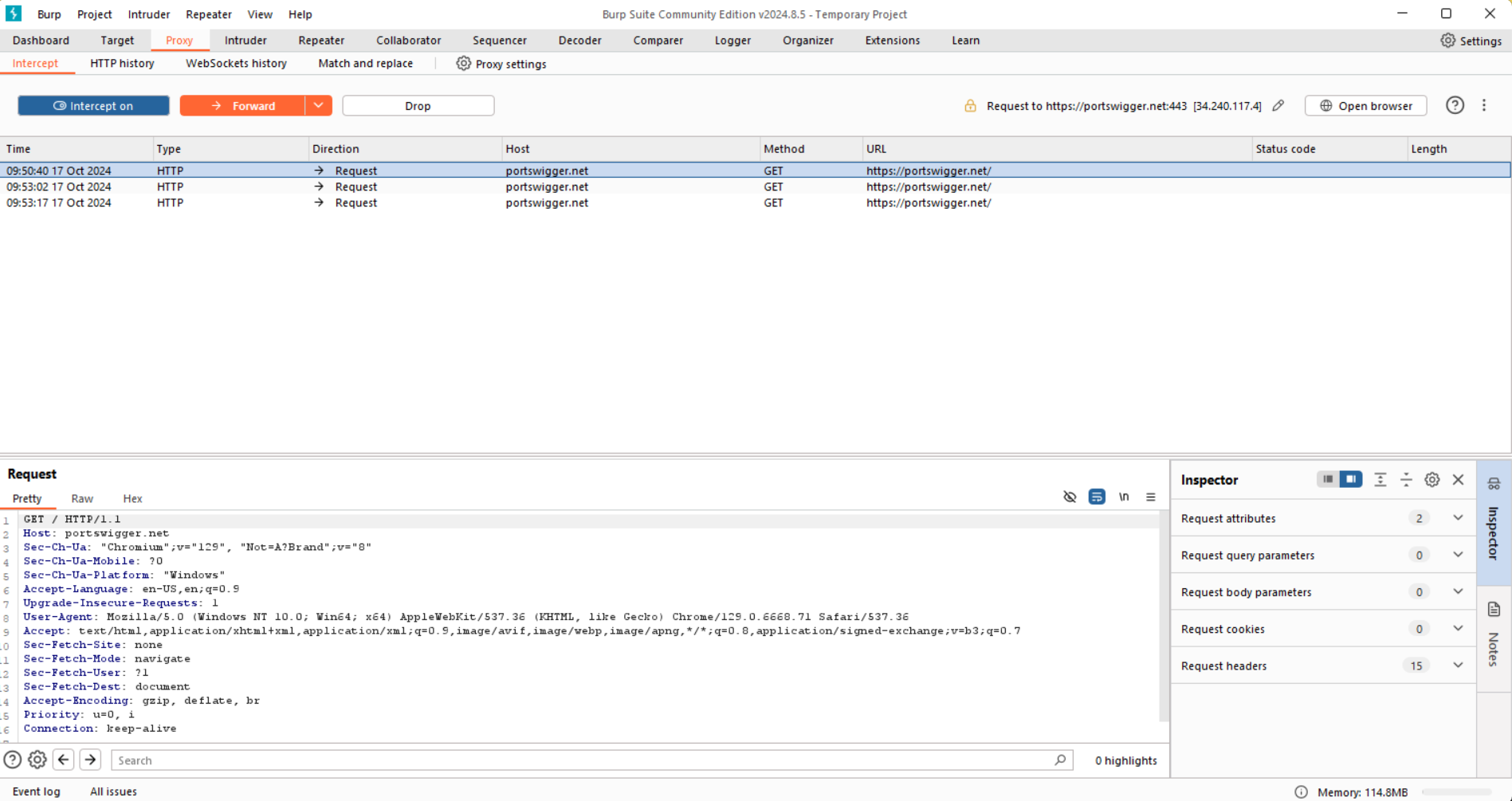
**Step 4: Click on proxy, turn intercept on and open browser**



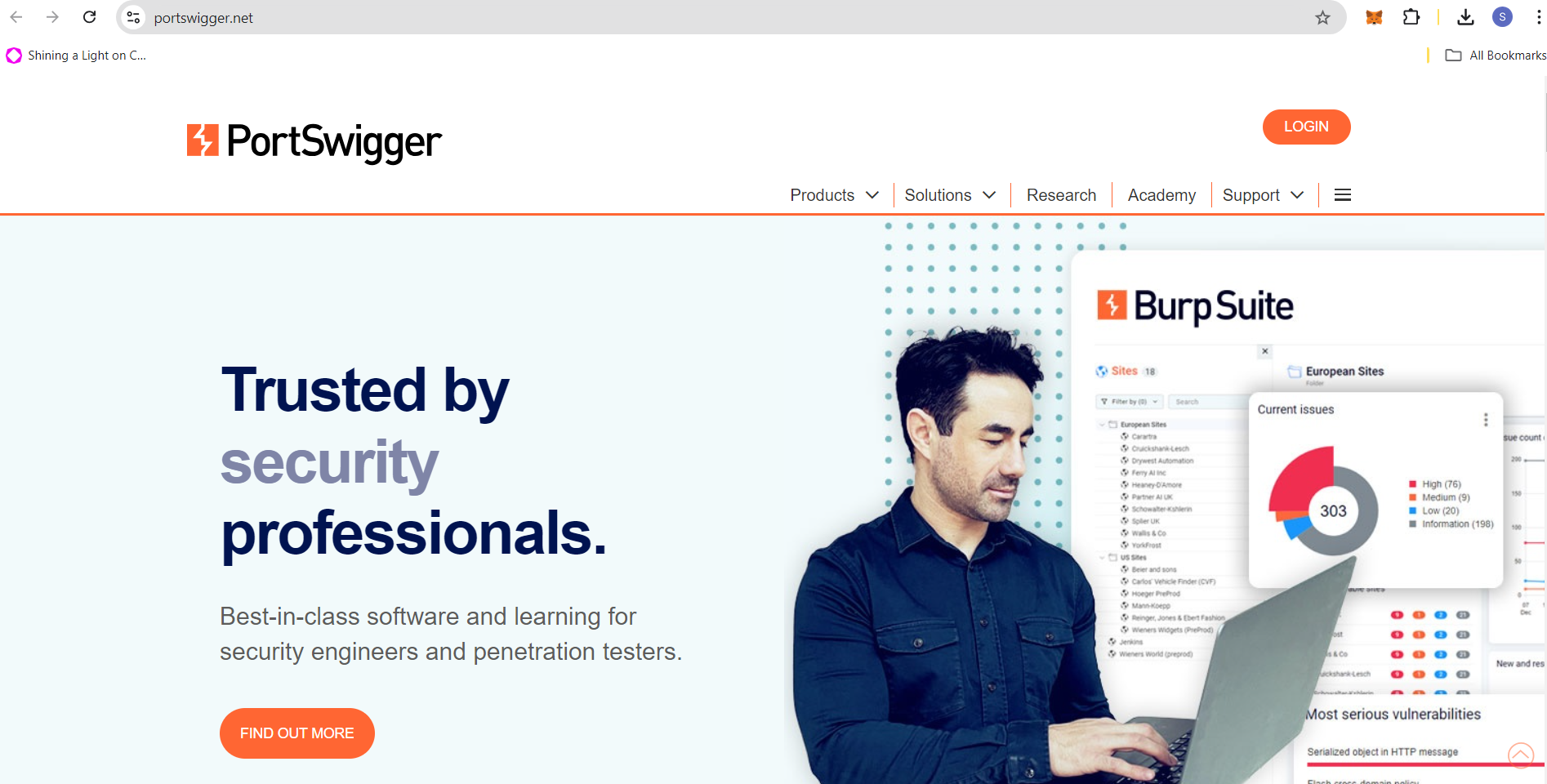
**Step 5 : Open https://portswigger.net/**



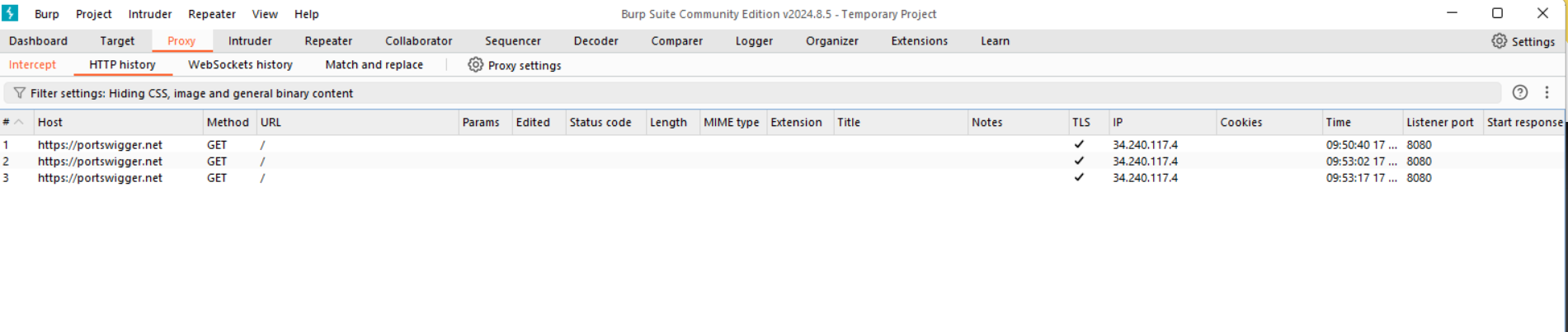
**Step 6 : We can now analyze and then forward the request to the server to load the page**

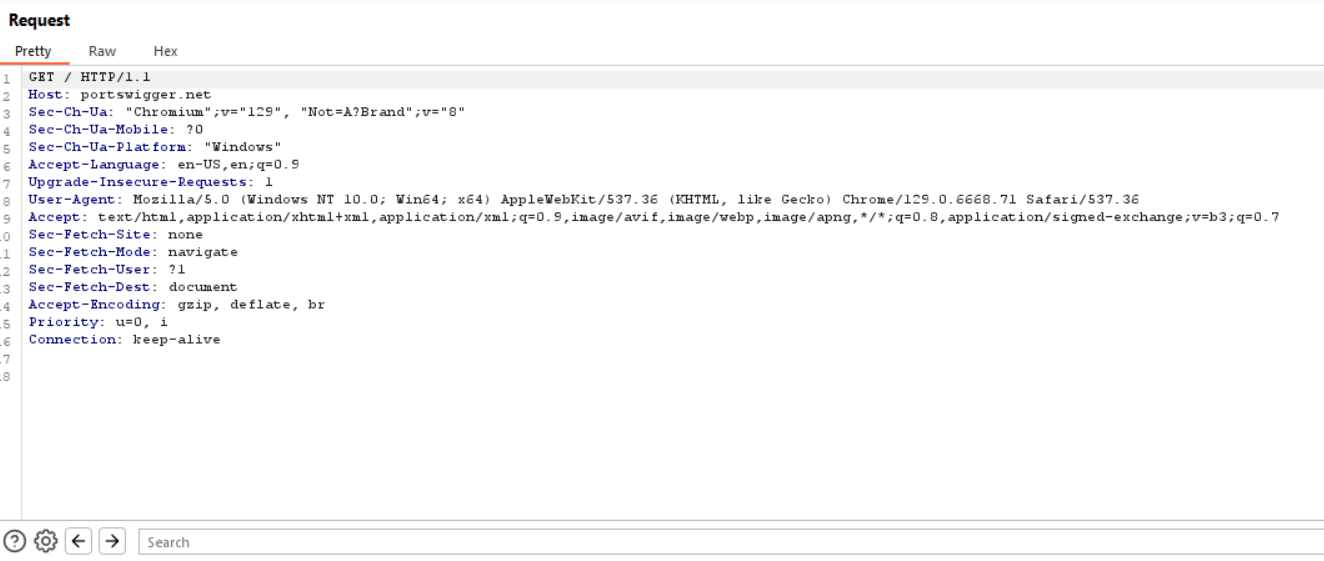


**Interception is done and the site has been forwarded to the server and the site has loaded in the user side.**



**Step 7: Go to HTTP History , where we can see all the logs of the user has been to even when intercept is off**





**Conclusion**:This we have used burp proxy to test the web applications.